vi. Software

The control system software is partitioned, according to functionality, into four functional layers:

- User Interface Layer
- Computer Services Layer
- Accelerator Device Object (ADO) Layer
- Equipment Control Layer

Each of the four layers performs specific functions, with well-defined interfaces to the layers immediately above and below it. The software interface between layers will be realized through procedure calls that may involve communications over local area networks. Standardized Remote Procedure Call (RPC) protocols will be used between the software processes that operate on the different computers of the control system. An object-oriented approach will be taken throughout the software development activities for the control system.

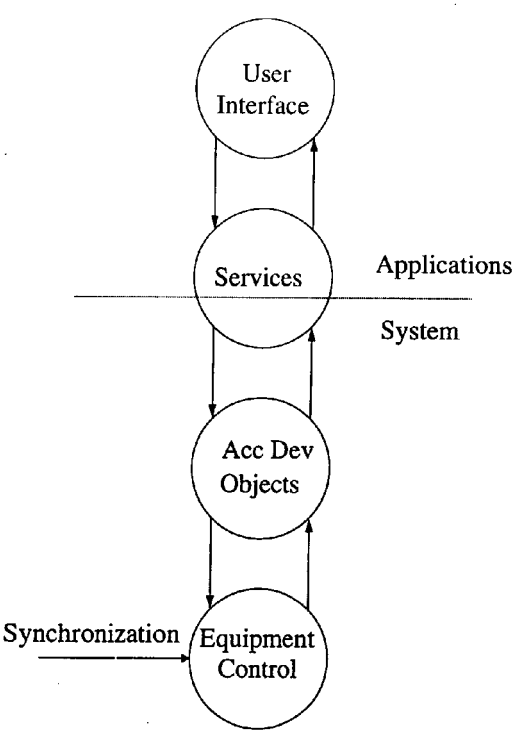

Fig. 9-2. Software Architecture

The lowest two layers (Equipment Control and ADO) will occupy the Front End Computer (FEC) level of the control system's physical architecture, operating under control of a real-time executive (VxWorks). Most of the functionality of the remaining two layers (Computer Services and User Interface) will reside in the console computer level. Applications programs, support tools, and other non-control-system software will also reside at the console computer level, under control of a UNIX operating system. The interrelations between these four functional layers is shown in Fig. 9-2. Class libraries will be used to enhance software reusability, and the design of the control system layers will be independent of I/O or network details.

Equipment Control Layer

This is the lowest layer in the distributed control system, and the only one which directly performs i/o to external devices and handles signals for events, clocks, data reference, etc. All control and data acquisition transactions with collider equipment take place in this layer or in the equipment itself. The equipment control layer will receive standardized messages (from the ADO layer) which contain requests for data or commands and setpoints for the equipment.

The equipment control layer is responsible for translating a standard data representation in the control system to a suitable format for devices and device control modules. It will perform the necessary conversions where analog control, analog-to-digital/digital-to-analog conversion is required by the equipment. For equipment with a digital interface, this layer is responsible for presenting data to the equipment in the proper digital form.

The equipment control layer forms the basic building block of the control system. Equipment under control, and associated control modules, each have unique designations in the control system as determined by location in the network and addressing in the FEC. Configuration data, including the equipment designations and operational parameters, are made available to this layer (and to the ADO layer) via tables within the memory of the FEC that are constructed from appropriate portions of a "Configuration Database".

ADO Layer

This layer comprises a set of Accelerator Device Objects (distributed among the FECs) that provide a high-level view of the collider equipment to the rest of the system. This view is independent of the details of the equipment control layer or the equipment itself. When the ADO layer responds to a request from the Computer Services Layer, raw data is transformed and formatted as necessary. These operations may include the conversion to engineering/physics units and the

conversion of computer addresses to and from user-friendly device names. A schematic representation of the organization of tasks in the FEC is given in Fig. 9-3.

From an "Object Oriented" perspective, an Accelerator Device Object is a class that contains a software data structure and the full set of actions or "methods" that are needed

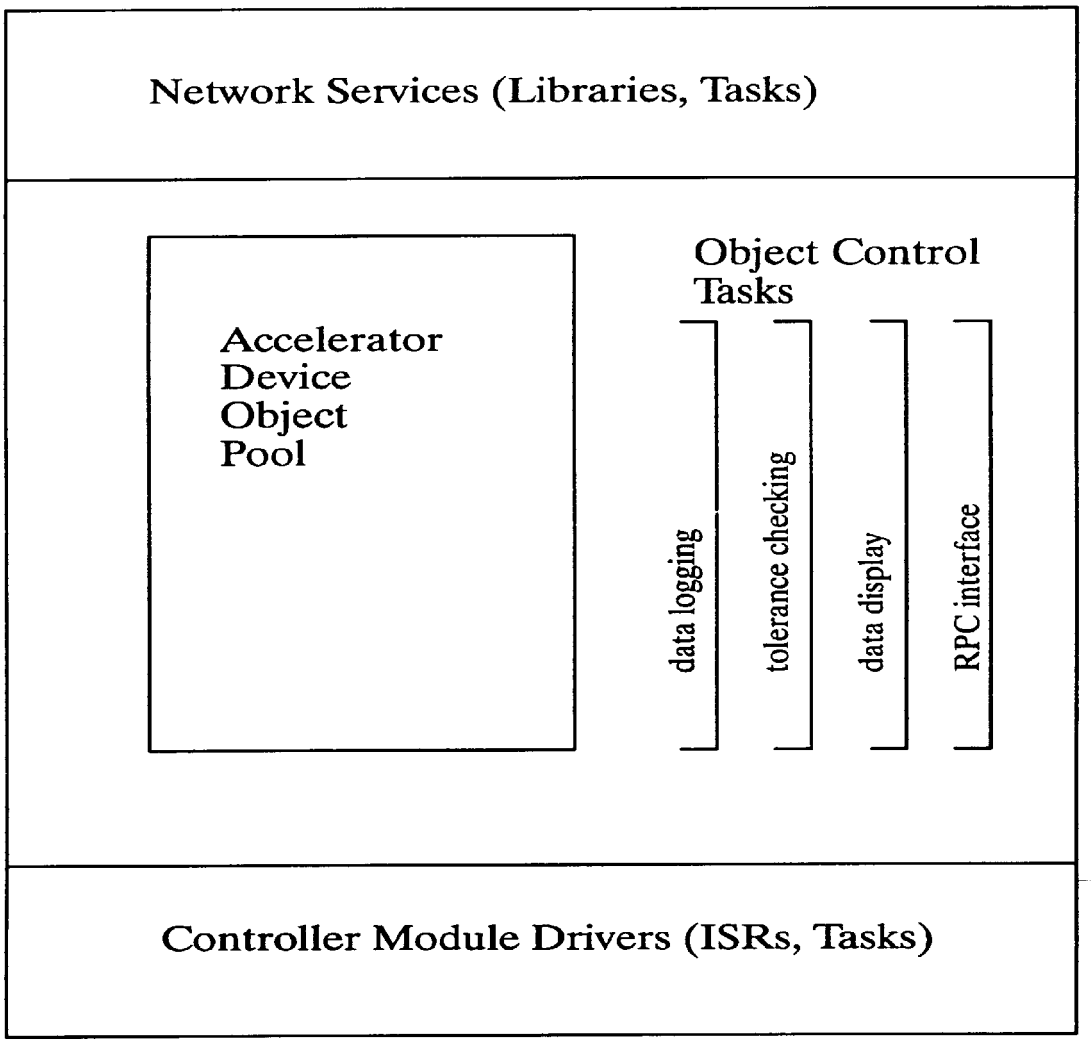

Fig. 9-3. FEC Task Organization

to control and monitor the accelerator device(s) which the particular ADO represents. The following is list of methods which an ADO may support:

create;	initialize;
reserve/unreserve;	get a measurement;
report data;	stream data/stop streaming;
set value(s);	watch a value or status/unwatch;
report history;	start logging/stop logging;
set tolerance(s);	set conversion parameters;
calibrate.	

The standard interface to the ADO layer is uniform for all subsystems. Settable parameters for an ADO may include modes and settings; readable data include status, measurements, mode readbacks, and setting readbacks.

Accelerator device objects in the control system will be "named". Resolution of human-assigned names into network addresses will be provided as a system service at the Computer Services Layer.

Computer Services Layer

The functions provided by this layer include those usually provided by any interactive, networked computer system, plus generic functions and special functions that are needed by control systems. Generic system functions which will be provided include:

- Communication protocol support (TCP/IP,UDP/IP,RPC)
- Computer security
- Network management tools (SNMP)
- Remote login (rsh,telnet)
- Network file systems (NFS,ftp)
- Printer support
- Remote X-terminal support
- X-client libraries

In addition to generic functions and custom applications programs, the following control system functions will be supported at this layer:

- Data viewing and analysis
- Message support (and a system of message categories)

- Relational database management (including equipment configuration database)
- Resolution of accelerator device names to network addresses
- Archiving and restoring of accelerator objects

This layer will also provide tools to support operational requirements, including page display, signal display, and timing diagnostics.

User Interface Layer

This layer provides the man-machine interface for control of the collider. It supports an interactive graphical user interface (GUI), as well as passive graphical and text displays and files for operations and analysis. Modern, user-friendly graphical input and output objects, such as pull-down and pop-up menus, menu bars, radio buttons, check lists, forms, text and graphical output, etc. will be supported in an X-window environment.